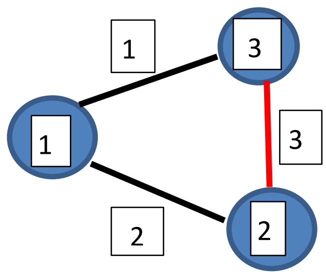
ניצן בר-אל 322450552 עילי כץ 212321871

עבודה מספר 3:

1)

א.



על פי האלגוריתם יוחזר כי העפ"מ האדום ביותר

הוא (2,3) , (1,3). משקלו הוא 4.

(1,3) , (1,2) הוא עץ פורש ומשקלו 3,

ולכן האלגוריתם החזיר עץ פורש לא מינימלי.

ב.א.

שלב 1: נמיין את צלעות E בסדר עולה לפי משקל (הקלה ביותר תופיע ראשונה, והכבדה תופיע בסוף), כך שלכל $v \in R, u \in E\backslash R$ כך ש w(v) = w(u) , v יופיע לפני u במערך הממוין, נסמן סידור זה ב A.

שלב 2: הרץ Kruskal על G לפי הסידור A והחזר את העץ המתקבל.

ב.ב.

האלגוריתם יפעל בדיוק כמו Kruskal ולכן יחזיר עפ"מ בוודאות. קיים מיון פנימי בין קשתות בעלות משקל שווה, זאת על מנת להעדיף אדומים. המיון יגרום לכך שלא תבחר קשת רגילה כל עוד נשארה קשת אפשרית לבחירה שהיא אדומה ובכך אכן יחזיר עפ"מ אדום ביותר.

ב.ג.

שלב 1: המיון – $O(|E|\log(|E|))$

שלב 2: Kruskal - $O(|E|\log(|V|))$,

כיוון והגרף קשיר $|E| \geq |V| - 1$, ולכן $O(|E|\log(|E|)) + O(|E|\log(|V|)) = O(|E|\log(|E|))$

לכן זמן הריצה הכולל של האלגוריתם הוא $O(|E|\log(|E|))$.

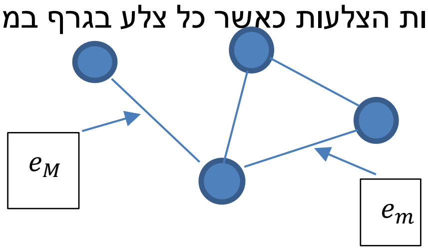
ניצן בר-אל 322450552 עילי כץ 212321871

(2)

א.

הפרכה

גרף G(V,E) קשיר ולא מכוון אשר קיימות הצלעות כאשר כל צלע בגרף במשקל שונה מהשנייה



e_M צלע בעל משקל מקסימאלי בגרף

e_m צלע בעל משקל מינמאלי בגרף

העץ הפורש המינימאלי של גרף זה יכיל את הצלע e_M לכן הפרכה לטענה

ב.

הוכחה

גרף G(V,E) קשיר ולא מכוון אשר קיימות הצלעותe_M צלע בעל משקל מקסימאלי בגרף

e_m צלע בעל משקל מינמאלי בגרף כאשר כל צלע בגרף במשקל שונה מהשנייה

נניח כי קיים עץ פורש מינימאלי $T(V_t, E_t)$ כך ש $e_m \notin E_t$ נבנה חתך מכבד של T אשר חותך את e_m וניצור עץ פורש חדש $T'(V_t, E_t')$ כאשר במקום הצלע e אשר נמצאת בT וגם בחתך נשים את e_m' במקומה כלומר $E_t' = (E_t \setminus e) \cup e_m$ המשקל של $w(T') = \sum w(e) \in E_t{}'$
המשקל של $w(T) = \sum w(e) \in E_t$

נראה עכשיו כי T עדיין עץ פורש מינמאלי $w(T') \setminus e_m$ = $w(T) \setminus e$ אבל $e_m < e$ ולכן $w(T') < w(T)$ סתירה להנחה שT עץ פורש מינימאלי

כלומר כדי שבגרף G שלנו יהיה עץ פורש מינימאלי T, T חייב להכיל את e_m.

ג.

הוכחה

נניח בשלילה כי קיימים לפחות 2 עפ"מים שונים ל G נקרא להם $T_1(V, E_1), T_2(V, E_2)$ נחלק ל3 מקרים :

מקרה 1 $\sum w(e_1) < \sum w(e_2)$ אזי T_2 איננו עפ"מ על G מכיוון שמצאנו עץ פורש קל יותר (T_1)

מקרה 2: $\sum w(e_1) > \sum w(e_2)$ אזי T_1 איננו עפ"מ על G מכיוון שמצאנו עץ פורש קל יותר (T_2)

מקרה 3:

$\sum w(e_1) = \sum w(e_2)$ כיוון ואין זוג צלעות בעל משקל שווה חייב להתקיים כי לכל

$\sum w(E_1 \setminus E_2) = \sum w(E_2 \setminus E_1)$ נגדיר את k להיות הצלע בעלת המשקל המינימאלי מקבוצת הצלעות $(E_1 \setminus E_2) \cup (E_2 \setminus E_1)$ נניח ש k נמצאת בE_1 אזי נוכל להחליף את k בצלע כלשהי בE_2 שעדיין T_2 יהיה עץ פורש. כלומר מצאנו עץ פורש קל יותר מT_2 (באופן סימטרי היינו עושים אם k הייתה נמצאת בE_2). לכן סתירה T_2 לא עפ"מ ולכן סתירה להנחה הראשונית שלנו שקיימים לפחות 2 עפ"מים על G.

3)

א.

$$G = \left[(A, d_A = 0), (X, d_x = 2), \left(Y, d_y = 7\right), (B, d_B = 11)\right], m = 10$$

קיימים שני פתרונות אופטימאלים לבעיה זו

1. *A,X,B*

2. *A,Y,B*

ב.

האלגוריתם

נמיין את מערך *G* לפי המרחקים *d* בסדר עולה כך שכל $d_i < d_{i+1}$. בהנחה שקיימות תחנות דלק כך שמרחקן מ-*A* גדול מהמרחק בין *A* ל- *B* נמחק אותן וזאת על ידי מציאת תחנה *B* והרצת האלגוריתם רק על תת המערך שמ-*A* ל-*B*

נבחר את המרחק d_i הגדול ביותר כך של $d_i \leq m + d_A, d_A = 0$

נחזור על הצעד כך שכל פעם נחפש המרחק הגדול ביותר שקטן מהנקודה בה אנו נמצאים ועוד *m*,

לכל נקודה *i* נחפש $\max(d_j)$ כך ש $d_j \leq d_i + m$ לכל *i < j* כך עד שנגיע ליעד *B*

ג.

הוכחת נכונות

יהי $G[1, \dots n]$ מערך תחנות הדלק לאורך הדרך כך ש d_i זה המרחק של תחנת הדלק במקום ה-*i*

מ-*A* , *m* זו כמות הדלק שהמיכל יכול להכיל.

יהי *O* פתרון אופטימאלי כלומר טיול מ*A* ל*B* בעל מספר מינימאלי של עצירות. *P* הפתרון שמחזיר האלגוריתם נראה כי $|O| = |P|$. נגדיר *P[i]* תת מערך של הפתרון *P* מ-*A* ועד לעצירה ה-*i* , נגדיר *O[i]* תת מערך של הפתרון *O* מ-*A* ועד לעצירה ה-*i*.

$$O[1] = [A] = P[1] \rightarrow |O[1]| = |P[1]|$$

זאת מכוון שכל פתרון חוקי חייב להכיל את *A* כיוון ומרחקו מ*A* מינימאלי הוא תמיד יהיה הראשון במערך.

יהי *i* מספר תחנת הדלק הראשונה כך ש *P[i-1] = O[i-1]* , $P[i] \neq O[i]$. נגדיר:

$G[t] \ni O[i] \backslash O[i-1]$, $G[k] \ni P[i] \backslash P[i-1]$.

כעת נגדיר $P[i] = O'[i] = O[i-1] \cup G[k]$, זהו עדיין תת מערך של פתרון אופטימלי כלשהו, כיוון ו- *P[i-1] = O[i-1]* אז בחירת *G[k]* כאיבר הבא הוא בחירה חוקית (על פי בחירת האלגוריתם). ובגלל דרך בחירת האלגוריתם כיוון וגם *G[t]* הינה בחירה חוקית אזי *k>t* . ולכן אם *O[i+1]* הוא תת מערך של פתרון חוקי אזי גם $(O[i+1] \setminus O[i]) \cup O'[i]$ הוא תת מערך של פתרון חוקי כיוון ו $d_k > d_t$ אז המרחק בין *G[k]* לאיבר האחרון ב *O[i+1]* קטן בוודאות מ m .

כך נשנה את הפתרון האופטימלי עד שנגיע ל *O' =P*. כיוון ובכל החלפה איננו מוסיפים איבר לפתרון האופטימלי אלא מחליפים אחד באחר, משמע $|O| = |P|$ כלומר *P* הוא פתרון אופטימלי כיוון והוא בעל מספר עצירות מינימלי.

ד.

מיון המערך $O(nlogn)$

מחיקת תחנות דלק מיותרות $O(n)$

בחירות תחנות עצירה (על ידי חיפוש בינארי) $O(nlogn)$

לכן סה"כ זמן הריצה $O(nlogn)$

4)

א.

$$P(u,v) = \{(v_1, v_2, \dots v_n) | v_1 = u, v_n = v, \forall i \in [1, n-1] \; (v_i, v_{i+1}) \in E\}$$

ב.

$$Forest(T) = \{T(V_T, E_T) | \forall v_i \in V_T \; \nexists P(v_i, v_i), \forall k, j \in [2, n-1], \; v_k, v_j \in P(v_i, v_i) \; v_k \neq v_j\}$$

ג.

$$Tree(T) = \{Forest(T) | \forall v_i, v_j \in V_T, \exists\, P(v_i, v_j)\}$$

ד.

$$ST_G(T) = \{Tree(T) | V_T = V\}$$

ה.

$$MST_G(T) = \{ST_G(T) | \; \forall (v,u) \in E_T, \left(\sum w(v,u)\right) \leq \left(\sum w(e)\right), e \in E_{T'} \;, \forall ST_G(T')\}$$

ו.

$$U - MST(T) = \{MST_G(T) | \nexists MST_G(T') \; E_{T'} \neq E_T\}$$

5)

א.

$$\forall x_i\ , \qquad e(x_i) \le e(x_s)$$

$$e(x_i) + e(x_w) \le e(x_w) + e(x_s) < P$$

הוכח לא קיים זוג גנראטורים שאחד מהם הוא x_w אשר עונה על הדרישה של לספק מספיק אנרגיה לכן לא קיים פתרון חוקי לבעיה אשר מכיל בתוכו זוג עם x_w

ב.

נחלק ל4 מקרים

מקרה 1: שניהם לא מצוותים כלל בSol_1

פתרון חוקי $e(x_w) + e(x_s) \ge P \rightarrow k = \{(x_s, x_w)\}$

על פי הגדרת Sol_1

$$k \nsubseteq Sol_1, \qquad Sol_2 = Sol_1 \cup k \rightarrow |Sol_2| > |Sol_1|$$

מקרה 2: x_w מצוות בSol_1 וx_s לא כך ש $(x_i, x_w) \in Sol_1$ כוון ו $e(x_s) \ge e(x_i)$ אזי

$$Sol_2 = Sol_1 \setminus \{(x_w, x_i)\} \cup \{(x_s, x_w)\} \rightarrow |Sol_2| = |Sol_1|$$

מקרה 3: x_w לא מצוות בSol_1 וx_s כן כך ש $(x_i, x_s) \in Sol_1$ כוון ונתון כי $e(x_w) + e(x_s) \ge P$

אזי

$$Sol_2 = Sol_1 \setminus \{(x_s, x_i)\} \cup \{(x_s, x_w)\} \rightarrow |Sol_2| = |Sol_1|$$

מקרה 4: x_w, x_s שניהם מצוותים ב Sol_1 אבל לא ביחד כלומר $(x_i, x_s), (x_j, x_w) \in Sol_1$

כיוון ו $e(x_i) \ge e(x_w)$ $e(x_w) + e(x_j) \ge P$ אז $e(x_i) + e(x_j) \ge P$ וגם נתון $e(x_w) + e(x_s) \ge P$

לכן

$$Sol_2 = Sol_1 \setminus \{(x_s, x_i), (x_w, x_j)\} \cup \{(x_s, x_w), (x_i, x_j)\} \rightarrow |Sol_2| = |Sol_1|$$

ג.

האלגוריתם:

נגדיר $G = \{\}$

נמיין למערך A את הגנראטורים על פי האנרגיה שהם מספקים בסדר עולה

נשמור מצביע לסוף המערך (הגנראטור אשר מפיק את האנרגיה הגבוהה ביותר) MAX
ומצביע לתחילת המערך (הגנראטור אשר מפיק את האנרגיה הנמוכה ביותר) MIN

נריץ בלולאה כל עוד MAX גדול מ- MIN:

אם e(A[MAX]) +e(A[MIN]) >= P נוסיף את הזוג הנ"ל לפתרון G , נשים את המצביע MAX על האיבר שלפניו במערך ($MAX = MAX - 1$).

נגדיל את MIN באחד ($MIN = MIN + 1$)

נחזיר את G

הוכחת נכונות:

נוכיח באינדוקציה כי $G \subseteq Sol_1$ (G הוא הפתרון שהאלגוריתם מחזיר בסוף ההרצה) :

בסיס האינדוקציה:

בצעד האפס: $G_0 = \{\}$ ולכן $G_0 \subseteq Sol_1$

הנחת האינדוקציה $Sol_1 \supseteq G_{i-1}$

צעד האינדוקציה :

$x_k.x_j$ הגנרטורים המקסימאלי והמינימאלי בהתאמה

נחלק למקרים:

אם $e(x_k) + e(x_j) < P$ לא נוסיף אותם ל G ולכן $G_i = G_{i-1} \subseteq Sol_1$

אחרת נוסיף אותם ל G $G_i = G_{i-1} \cup \{(x_j, x_k)\}$

מ5 סעיף ב' קיים $(x_j, x_k) \in Sol_2$ כך ש Sol_2 פתרון חוקי וגם $|Sol_2| \geq |Sol_1|$

$Sol_2 \supseteq G_{i-1}$ כיוון ו $G_{i-1} \notin (x_j, x_k)$ אז לא הורדנו אף זוג מG_{i-1} בבניית Sol_2 וכיוון ו

$(x_j, x_k) \ni Sol_2$ אז $Sol_2 \supseteq G_i$. ולכן הוכחנו שG_i מוכל בפתרון אופטימאלי כלשהו.

בזכות האינדוקציה ניתן לומר כי G מוכל בSol_1 כלשהו.

נניח בשלילה כי G אינו פתרון אופטימאלי כלומר קיים $Sol_1 \setminus G \in (x_y, x_z)$ כך ש x_y or x_z לא נכנסו לG בשום ציוות. נלך לאיטרציה באלגוריתם שבו נבחר x_y (הגנרטור שלא צוות בG מבין השניים, במידה ורק x_z לא צוות נתייחס לx_y כאל x_z) אזי x_y הינו *A[MIN]* ו+ $e(x_y)$ $P > (A[Max])e$ועל פי 5 סעיף א' לא קיים ציוות חוקי עם x_y . ל$Sol_1 \notin (x_y, x_z)$ מכאן סתירה להנחה.

ולכן G הינו פתרון אופטימאלי